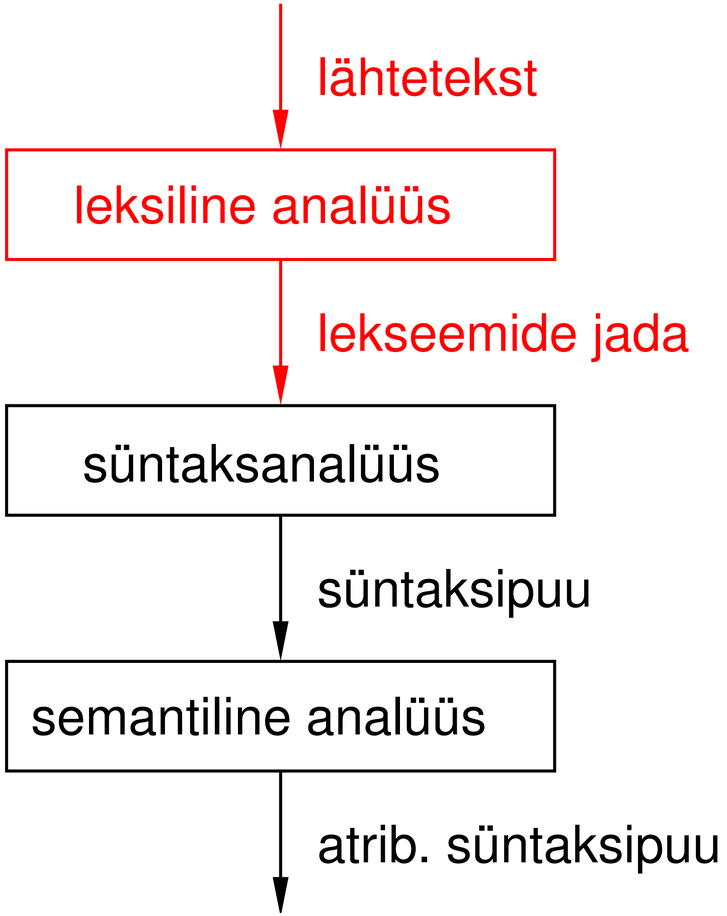
Automaadid, keeled, translaatorid

Leksiline analüüs
Regulaaravaldised

Leksiline analüüs



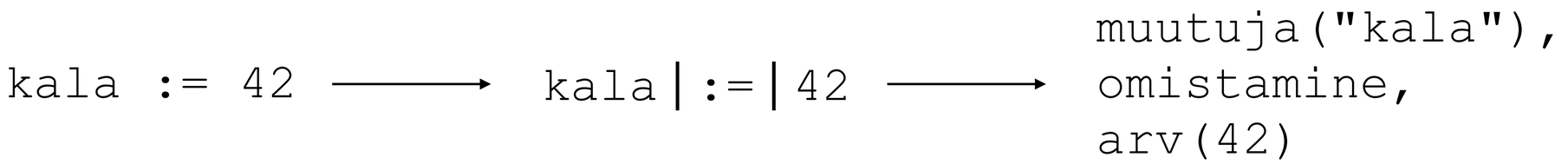

```
x := y + 60
```



```
muutuja(x),
  omistamine,
    muutuja(y),
      liitmine,
        arv(60)
```

Lekser ja skanner

- Leksiline analüsaator (lekser) tükeldab sisendit ja loob lekseeme.



- Sõnade piiride leidmist nimetatakse ka skannimiseks.

- Lisaks on lekseemide loomisel viisakas konstante väärtustada.
  Näiteks: Kui „4“ ja „2“ on kõrvuti, siis väärtuseks on 4*10+2 = 42.

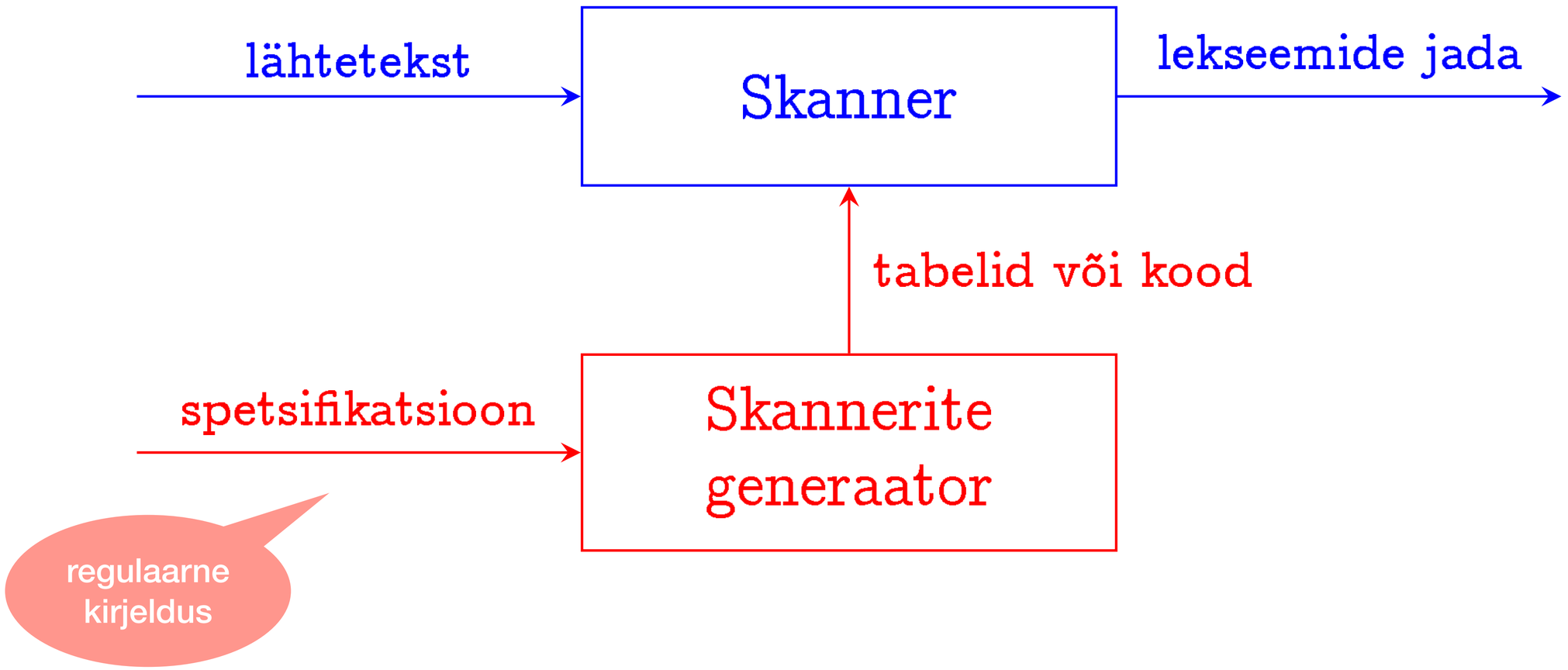
- Rangelt võttes võib neid samme eristada: lekser = skanner + väärtustaja.

Tihti aga kasutatakse sõnu lekser ja skanner lihtsalt sünonüümidena.

Käsitsi kodeeritud skanner

- Paljude keelte lekserid on käsitsi kirjutatud (sealhulgas OpenJDK).
- Nagu saame näha (5. kodutöös), siis on see üsna raske.
- Meie kasutame parema meelege lekseri generaatorit,
  mis loob lekseri automaatselt regulaaravaldiste põhjal.

Skannerite generaator



Näiteid regulaarsetest kirjeldustest

Identifikaatorid:

$$
\begin{array}{lcl}
\text{Letter} & \to & [\texttt{a} - \texttt{z}\ \texttt{A} - \texttt{Z}] \\
\text{Digit} & \to & [\texttt{0} - \texttt{9}] \\
\text{Identifier} & \to & \text{Letter}\,(\text{Letter} \mid \text{Digit})^{\star}
\end{array}
$$

Arvkonstandid:

$$
\begin{array}{lcl}
\text{Sign} & \to & (\texttt{+} \mid \texttt{-})? \\
\text{Integer} & \to & \texttt{0} \mid \text{Sign}\,[1 - 9]\,\text{Digit}^{\star} \\
\text{Decimal} & \to & \text{Integer}\ \texttt{.}\ \text{Digit}^{+} \\
\text{Real} & \to & (\text{Integer} \mid \text{Decimal})\,\texttt{E}\,\text{Integer}
\end{array}
$$

Formaalne keel

- Vajame mugavat vahendit, millega defineerida sõnade hulkasid.
- Olgu meil lõplik hulk tähti Σ (tähestik).
  Näiteks: eesti tähestik või hulk {0,1}.
- Kui võtta null või enam tähte hulgast Σ ja need kõrvuti paigutada,
  siis moodustub sõna! (Näiteks *kala* ja *1011001*).
- Σ^* tähistatakse kõikide sõnade hulka üle tähestiku Σ.
- Formaalne keel on suvaline Σ^* alamhulk!
  Regulaaravaldiste abil saabki keelt defineerida (siis on ta regulaarne keel).

Regulaaravaldised

Avaldis	Nimi	Kirjeldus	Näide
a	Literaal	Üksiktäht meie tähestikust	“a”
ε	Tühisõna	Epsilon tähistab tühja sõna	“”
s | t	Alternatsioon	Valik avaldiste sõnade vahel	“s” ja “t”
s t	Konkatenatsioon	Esimese avaldise sõna ja siis teise sõna	“st”
s*	Kleene’i tärn	Avaldise kordus	“”, “s”, “ss”, “sss”, …

Ja neid saab muidugi kombineerida: avaldise (a|b)c keelde kuulub nii “ac” kui ka “bc”

Regulaaravaldise prioriteedid

- Nagu ka aritmeetikas, võib regulaaravaldises sulud ära jätta!
- Näiteks kirjutame avaldise `(a*(b|ε))|((bc)a)` asemel lihtsalt `a*(b|ε)|bca`.
- Operaatorite prioriteedid on kahenavas järjekorras kordus, konkatenatsioon ja alternatsioon.

Mata	AKT
a^n	a*
a × b	ab
a + b	a | b

Keele formaalne definitsioon

$$
\begin{aligned}
L(\varepsilon) &= \{\text{""}\} \\
L(\mathrm{a}) &= \{\text{"a"}\} \\
L(E_1\ E_2) &= \{vw \mid v \in L(E_1),\ w \in L(E_2)\} \\
L(E_1 \mid E_2) &= L(E_1) \cup L(E_2) \\
L(E^\star) &= \{\text{""}\} \cup \{vw \mid v \in L(E), w \in L(E^\star)\}
\end{aligned}
$$

Näited

Regulaaravaldis	Defineeritav keel
$a \mid b$	{"a", "b"}
abba	{"abba"}
$(a \mid b)(c \mid d)$	{"ac", "ad", "bc", "bd"}
$ab^\star a$	{"aa", "aba", "abba", "abbba", ...}
$(ab)^\star$	{"", "ab", "abab", "ababab", ...}
$(a \mid b)^\star$	{"", "a", "b", "aa", "ab", "ba", "aaa", "aab", "aba", "baa", "abb", "bab", "bba", "bbb", ...}

Kokkuvõte

- Leksiline analüüs
- Leksiline analüsaator ehk lekser ning selle genereerimine
- Formaalne keel
- Regulaaravaldised, nende koostamine ja nendele vastavad keeled